

CoCoA-5 - Support #879

LOGAR: make CoCoA-5 easier to use

09 May 2016 15:05 - John Abbott

Status:	New	Start date:	09 May 2016
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	enhancing/improving	Estimated time:	0.00 hour
Target version:	CoCoA-5.?.?	Spent time:	1.50 hour
Description			
<p>In early May 2016 I met (Sandro) Logar at the Normaliz workshop in Osnabrueck. I wanted to persuade him to use CoCoA-5, so we developed together a small session aiming to solve one problem of the sort he usually deals with.</p> <p>The computation he wanted to do was not so difficult, but it was surprisingly awkward to do it with CoCoA-5.</p> <p>We should try to extended CoCoA-5 so that it is easier to use for this sort of calculation.</p>			
Related issues:			
Related to CoCoA-5 - Bug #880: subst should check that indet is in same ring ...		Closed	09 May 2016

History

#1 - 09 May 2016 15:10 - John Abbott

```
// The problem is to find rational points on an ellipse.
// We work with a homogeneous ellipse (hence indets x,y,z).
// The ellipse passes through the origin: proj point (0,0,1).
// We intersect the ellipse with a line passing through the
// origin given by alpha*x+beta*y=0. The idea is that
// alpha/beta will be rational (if beta is non-zero).
// We want to deduce expressions for the coordinates of the
// other point where the line meets the ellipse.

QQa1 := QQ[a[1..6], alpha, beta];
QQa := NewFractionField(QQa1);
P := QQa[x,y,z], lex;
use P;

conica := ScalarProduct(support((x+y+z)^2), [a[i] | i in 1..6]);

cond1 := subst(conica, [[x,0], [y, 0], [z, 1]]);

-- GIVES ERROR !conical := subst(conica, a[6], 0);
conical := sum(first(monomials(conica), 5));

retta := alpha*x + beta*y;
//retta;

I := ideal(conical, retta);
RGB := ReducedGBasis(I);
indent(RGB);

RGB1 := RGB[1];
C := coefficients(RGB1);
CommonD := lcm([den(c) | c in C]);
RGB1 := CoeffEmbeddingHom(P) (EmbeddingHom(QQa) (CommonD)) * RGB1;

P2 ::= QQ[a[1..6], alpha, beta, x, y, z];
use P2;
images := concat([a[i] | i in 1..6], [alpha, beta]);
CoeffHom := InducedHom(QQa, PolyAlgebraHom(QQa1, P2, images));

phi := PolyRingHom(P, P2, CoeffHom, [x,y,z]);

facs := factor(phi(RGB1));
```

```
indent (facs.factors);
```

#2 - 09 May 2016 15:16 - John Abbott

cond1 corresponds to the condition that the ellipse passes through the origin.
It simplifies to $a[6]=0$.
We would now like to add this condition to the coeff ring of parameters.

Given how simple the condition is the simplest approach is just to substitute $a[6]=0$ but the command subst does not let us because $a[6]$ is not an indet of the poly ring P.

More generally, the condition would be a polynomial, and we would probably want to replace the coeff ring by a quotient ring; or actually the fraction field of the quotient ring... but CoCoALib will not let us create a fraction field of a quotient ring (because it wants an explicit GCD domain).

In the end I had to resort to a dirty *ad hoc* trick to eliminate the term containing $a[6]$; this obviously does not extend to more complicated situations.

#3 - 09 May 2016 15:25 - John Abbott

Sandro then wanted to compute the expressions for the coordinates of the intersection points of the line and the ellipse. This was easy with a lex GBasis.

One of the expressions was a rational function. He now wanted to factorize something (now I don't recall what). Anyway, it seemed necessary to clear denominators of the polynomial with rational function coeffs, and then try to factorize that.

Clearing denoms was not too hard, but it is definitely tedious having to re-embed the common denom into the poly ring just so we can multiply. The result is visibly a polynomial, but it cannot be factorized because the coeffs are in a fraction field.

In fact, to factorize the result it has to be mapped into a polynomial ring of the form $QQ[\dots]$ which is what I did in the last (rather tedious) part. All in all, I am not sure that CoCoA-5 made much of a good impression :-)

Also note that the InducedHom from the fraction field is only a partial homomorphism: it will obviously fail on any value whose denominator maps to 0. This is strictly a bug in the design of CoCoALib... it can be fixed by inducing a hom from a subring which is a localization away from the kernel of original homomorphism...

#4 - 09 May 2016 15:30 - John Abbott

At some point I triggered a bug in CoCoA-5. I was trying to use subst somehow (to replace $a[6]$ by zero); perhaps it was on an image of conica in a purely polynomial ring? Anyway, it produced an "index out of range" error: the given index was 6, the max allowed was 3. I note that $a[6]$ has indet index 6, and that the poly ring P has max indet index 3. So maybe the code does not check that the first arg and the indet are in the same ring?

NOTE confirmed the bug: after the definition of cond1 the following call `subst(conica, RingElem(QQa1,["a",6]),0)` produces an error in SubstApply at line 580 of misc.cpkg5 saying
--> ERROR: The index 6 is not in the range 1..3

#5 - 09 May 2016 17:00 - John Abbott

- Related to Bug #880: subst should check that indet is in same ring as 1st arg, but does not. added